

Unified Interleaver/De-interleaver

FIELD OF THE INVENTION

[0001] The present invention relates generally to wireless communication systems.

BACKGROUND

[0002] In wireless communication systems, block interleaving may be used to combat channel fading. Generally, interleaving re-orders bits of information to achieve time diversity of an original sequence of bits (*i.e.*, interleaving changes the order of at least some bits in an original sequence with respect to time). When a transmitted interleaved signal is received, de-interleaving may be used to re-order the transmitted sequence back to the original sequence.

[0003] A variety of interleaving schemes exist in different wireless standards. Even in the same standard, there may still be a variety of coding schemes that use different interleaving algorithms. For example, the GSM/GPRS/EDGE standard has more than 15 different interleaving algorithms, including various look-up tables (LUTs). Interleaving algorithms may vary in block size (*i.e.*, the amount of bits in an index), level of interleaving, use of bit pruning, and other considerations as will later be described.

[0004] Most interleaver/de-interleaver implementations may typically be tailored for use with a specified algorithm and may not easily be used with other algorithms. Being able to implement a plurality of different algorithms with one

hardware and/or software solution is desirable. Furthermore, finding a multi-standard interleaving solution that does not require large amounts of memory and/or complex hardware is desirable.

SUMMARY

[0005] An interleaver/de-interleaver that may be used for multiple interleaving algorithms and look up tables (LUTs) of one or more interleaving standards. In at least some embodiments, the interleaver/de-interleaver may comprise an initial value selector, offset selector, and a pruning adjuster coupled to a combining block. The interleaver/de-interleaver may further comprise a boundary regulator coupled to the combining block, wherein the boundary regulator is configurable to modify an output of the combining block according to one or more pre-determined rules. The interleaver/de-interleaver may further comprise a controller coupled to, at least, the initial value selector, the offset value selector, and the pruning adjuster, whereby the interleaver/de-interleaver may interleave or de-interleave in accordance with a plurality of interleaving/de-interleaving techniques.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] For a detailed description of various embodiments of the invention, reference will now be made to the accompanying drawings in which:

Figure 1 illustrates a system for interleaving and/or de-interleaving in accordance with embodiments of the invention;

Figure 2 illustrates another system for interleaving and/or de-interleaving in accordance with embodiments of the invention;

Figure 3 illustrates a table of parameters for implementing an interleaving/de-interleaving algorithm using the system of Figure 2;

Figures 4A and 4B illustrate software code for implementing an interleaver/de-interleaver in accordance with embodiments of the invention;

Figures 5A-5S illustrate tables of parameters and parameter values that may be used to implement algorithms found in the GSM 05.03 V8.5.0 Release 1999 standard in accordance with embodiments of the invention;

Figures 6A-6B illustrate an interleaving algorithm (intra frame interleaving) according to a WCDMA standard found in 3GPP TS 25.212-v.3.5.0 (2000-12);

Figures 6C-6D illustrate tables of parameters and parameter values that may be used to implement the WCDMA standard of Figures 6A-6B in accordance with embodiments of the invention;

Figures 6E-6F illustrate a 30x30 look-up table (LUT) that may be used to implement the de-interleaver of the WCDMA standard of Figures 6A-6B in accordance with embodiments of the invention;

Figures 7A-7E illustrate tables of parameters and parameter values that may be used to implement algorithms of the IS2000 standard referenced in 3GPP2 C.S0002-C Version 1.0 in accordance with embodiments of the invention;

Figure 8 illustrates a system that implements an interleaver/de-interleaver in accordance with embodiments of the invention; and

Figure 9 illustrates a method for interleaving/de-interleaving data in accordance with embodiments of the invention.

[0007] While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

NOTATION AND NOMENCLATURE

[0008] Certain terms are used throughout the following description and claims to refer to particular system components. As one skilled in the art will appreciate, companies may refer to a component by different names. This document does not intend to distinguish between components that differ in name but not function. In the following discussion and in the claims, the terms “including” and “comprising” are used in an open-ended fashion, and thus should be interpreted to mean “including, but not limited to...”. Also, the term “couple” or “couples” is intended to mean either an indirect or direct electrical connection. Thus, if a first device couples to a second device, that connection may be through a direct electrical connection, or through an indirect electrical connection via other devices and connections.

DETAILED DESCRIPTION

[0009]The subject matter disclosed herein generally relates to wireless communication systems. As previously mentioned, block interleaving may be used in wireless communications to combat channel fading. An interleaver (or de-interleaver) may be described as a look-up table ("LUT") or a set of equations where mods (*i.e.*, modulus functions), multipliers, and dividers are used so that an arbitrary index can be mapped to its interleaved index.

[0010]As an example, consider wireless voice transmission (*e.g.*, cell phones). At a first cell phone, the voice of a user may be processed as blocks of data. The blocks of data and/or bits within each block may be interleaved, such that bits of data that were ordered consecutively are now separated by other bits. The interleaved blocks of data may be transmitted by the first cell phone and received by a second cell phone. At the second (receiving) cell phone, the blocks of data may be de-interleaved (*i.e.*, the original data sequence is restored) so that a user of the second cell phone may hear the voice characterized by the original blocks of data. The interleaving and de-interleaving process may reduce fading effects during wireless transmission of the voice data by spreading out interference. In addition to interleaving/de-interleaving, other measures (*e.g.*, modulation, encoding) may be used to reduce channel fading.

[0011]In some embodiments, the function of interleavers and de-interleavers may be described as a non-linear differential equation:

$$\text{index}(i+1) = \text{index}(i) + f'(i, \text{other_parameters}), \quad i=0,1,\dots,N-1 \quad (1)$$

where N is the interleaver/de-interleaver block length, $\text{index} ()$ is the interleaved index, and f' is a non-linear function that provides the distance (Δ) for two

consecutive input bits. For implementation convenience this equation may be rewritten as:

$$\text{index}(i+1) = \text{index}(0) + f(i, \text{other_parameters}), \quad i=0,1,\dots,N-1 \quad (2)$$

[0012] where N is the interleaver/de-interleaver block length, $\text{index}(0)$ is the first interleaved value of the original index, and f is a non-linear function that provides the distance (delta) for two consecutive input bits. The function $f(i, \text{other_parameters})$ may be decomposed into two parts: an offset part and an adjustment part. The offset part may calculate an offset relative to the initial value “ $\text{index}(0)$ ”, while the adjustment part may change the value of the calculated offset in case the calculated offset is not desired (e.g., in bit pruning cases).

[0013] Bit pruning removes unwanted bits of data from a data stream (e.g., a data stream associated with an interleaver algorithm) and compensates for spaces that are left in a data stream after unwanted bits have been removed. For example, a data block of arbitrary length may have a number of unrelated control bits that have been inserted between data bits of the data block. When de-interleaving, the control bits need to be taken out of the data block to prevent corruption of the data. As another example, in the WCDMA (Wideband Code Division Multiple Access) standard, an interleaving algorithm may write to a $N \times 30$ array, where N is the number of rows and 30 is the fixed number of columns. Because the data block length may be arbitrary, the last row of this array may not have exactly 30 elements. Therefore, there may be some blank bits in the last

row. Consequently, when an interleaver reads the array column by column, those blank bits need be pruned in the interleaving algorithm.

[0014]As previously mentioned, the function f may be non-linear and may comprise modulus, multiplication, and division operations, or alternatively may comprise an LUT that provides the offset distances with respect to an index origin, e.g., index (0). In order to simplify implementation of an interleaver/de-interleaver, embodiments of the invention may constrain the input or output of the index to be linear (*i.e.*, linearly increasing). For example, if an input to an interleaver is linear, the output may be a “randomly” indexed output sequence. Alternatively, if an output of an interleaver is linear, the input may be “random”. This constraint satisfies operational environments in which a block of data is processed bit by bit as an input or an output.

[0015]By using linear input or output indices, embodiments of the invention may replace multiplier(s), modulus operator(s), and divider(s) commonly used for function f with accumulator(s), subtractor(s), and counters, respectively. In some embodiments, the elements of equation (2) may be vectorized, *i.e.*, the initial index value “index (0)”, the mapped index “index($i+1$)”, and the components of the f function are vectors wherein each element of the initial vectors may be viewed as the first value of each column in a matrix. Additionally, the offset part of f may be unified as a vectorized module operator “ $mod(k\Delta, c)$ ”, where k is a linearly increasing auxiliary index, Δ is a difference vector, and c is a constant module value. In such embodiments, only accumulator(s) and subtractor(s) may be needed to implement the $mod(k\Delta, c)$ operation because of the linearity of k .

[0016]Figure 1 illustrates a block diagram of an interleaving/de-interleaving system 100 in accordance with embodiments of the invention. As shown in Figure 1, the system 100 may comprise an initial value selector/generator 102, an offset selector 104, a pruning adjuster 106, a combining block 108, a controller 112, and a boundary regulator 114.

[0017]As shown in Figure 1, the initial value selector/generator 102 may couple to the controller 112 and the combining block 108. The offset selector 104 may couple to the controller 112, the pruning adjuster 106, and the combining block 108. The pruning adjuster 106 may couple to the controller 112, the offset selector 104, and the combining block 108. The combining block 108 may couple to the initial value selector/generator 102, the offset selector 104, the pruning adjuster 106, the controller 112, and the boundary regulator 114. The boundary regulator 114 may couple to the combining block 108, the controller 112, and an address calculator 116. The address calculator 116 may also couple to the controller 112.

[0018]As illustrated in Figure 1, the initial value selector/generator 102 may receive as input an initial vector having one or more values. Additionally, the initial vector may comprise a plurality of vectors. In at least some embodiments, the initial vector may comprise one or more predetermined vectors that may correspond to a particular algorithm and/or whether the system 100 is used for interleaving or de-interleaving. Additionally, or alternatively, the initial vector may be programmable. In operation, the initial value selector/generator 102 may select an initial vector value and output that value to the combining block 108. In

at least some embodiments, a control signal 122 from the controller 112 may determine which initial vector value is selected and output to the combining block 108. Additionally or alternatively, the initial value selector/generator 102 may generate initial values. Therefore, in at least some embodiments an initial value vectors may not be used.

[0019] The offset selector 104 may receive as input an offset vector having one or more values. In at least some embodiments, the offset vector may comprise a predetermined vector that corresponds to a particular interleaving algorithm and/or whether the system 100 is used for interleaving or de-interleaving. Additionally, or alternatively, the offset vector may be programmable. In operation, the offset selector 104 may select an offset vector value and output that value to the combining block 108. In at least some embodiments, a control signal 124 from the controller 112 may determine which offset vector value is selected and output from the offset selector 104 to the combining block 108. Additionally, the offset selector 104 may receive an input from the pruning adjuster 106, whereby an adjusted offset value is output to the combining block 108.

[0020] The pruning adjuster 106 may change (*i.e.*, adjust) the value of the offset value received by the offset selector 104. In at least some embodiments, the pruning adjuster 106 function according to a control signal 126 from the controller 112. For example, the pruning adjuster 106 may adjust the offset value by adding or subtracting an amount determined by the control signal 126. Additionally, or alternatively, the pruning adjuster 106 may change add or

subtract a value from the combining block 108 operation according to the control signal 128 as will later be explained.

[0021] For example, there may be interleaving/de-interleaving algorithms in which using a fixed offset value is undesirable (such as when bit pruning is used). Accordingly, in at least some embodiments, the pruning adjuster 106 in coordination with the controller 112 may implement a bit pruning mechanism as previously described. Additionally, or alternatively, the pruning adjuster 106 may be used to account for interleaving algorithms that implement “burst mapping” (e.g., at least some algorithms in the GSM standard implement burst mapping). Burst mapping may comprise another level of interleaving (e.g., block diagonal interleaving in the GSM standard). In some embodiments, each burst may comprise a number of interleaved data blocks with bits from different data blocks ordered consecutively.

[0022] The combining block 108 receives an output value from the initial value selector/generator 102, the offset selector 104, and the pruning adjuster 106. By combining these outputs, the combining block 108 creates an “offset index position” that may be used to interleave or de-interleave a single bit of an index of bits. As the name infers, the offset index position may be an index position that is offset (*i.e.*, separated) from some original or “base” index position. The base index position may be a predetermined starting address of a block of data (e.g., “index 0” as described previously). In some embodiments, the base index position may be a previous offset index position.

[0023] The offset index position may be received by a boundary regulator 114, which functions to output an index position within the boundaries of a predetermined index. In some embodiments, the boundary regulator may determine if the offset index position is within the index boundary of a predetermined index. If the offset index position is within the index boundary, that offset index position may be output from system 100 for use with interleaving or de-interleaving a block of data. If the offset index position is not within the index boundary, an adjustment may be made so that the offset index position is modified to be within the index boundary. In some embodiments, one or more pre-determined rules may be used to modify an offset index position when necessary. For example, a predetermined index number (*i.e.*, amount) may be subtracted from the offset index position so that the offset index position is moved to within the boundaries of the index (*i.e.*, a modulus operation may be performed). In at least some embodiments, the amount subtracted from the offset index position may be equal to the data block size. More specifically, if an index [0:455] is to be interleaved, then an amount of 456 may be subtracted from an offset index position that is not within the [0:455] boundary. Assuming that the resultant index position is generated by an addition of an initial index value and a positive offset, the offset index position may possibly exceed the upper boundary of an index. A more detailed example will later be described. The output of the boundary regulator 114 may be used by an address calculator 116 to interleave or de-interleave a block of data. The address calculator 116 may also be a

vector of multiple addresses wherein different indexes may be combined with different base addresses.

[0024] Figure 2 illustrates another embodiment of an interleaving/de-interleaving system 101 in accordance with embodiments of the invention. As shown in Figure 2, the system 101 may comprise an initial value selector/generator 102, an offset selector 104, an pruning adjuster 106, a combining block 108, a controller 112, and a boundary regulator 114. The initial value selector/generator 102 may couple to the controller 112 and the combining block 108. The offset selector 104 may couple to the controller 112, the pruning adjuster 106, and the combining block 108. The pruning adjuster 106 may couple to the controller 112, the offset selector 104, and the combining block 108. The combining block 108 may couple to the initial value selector/generator 102, the offset selector 104, the pruning adjuster 106, the controller 112, and the boundary regulator 114. The boundary regulator 114 may couple to the combining block 108, the controller 112, and an address calculator 116. The address calculator 116 may also couple to the controller 112.

[0025] As shown in Figure 2, the initial value selector/generator 102 may comprise a multiplexer 134 that couples to another multiplexer 132 and an initial value generator 136. The offset selector may comprise a multiplexer 142 that couples to one or more accumulator/subtractors (ACSS) 144. The outputs of the ACSSs 144 may couple to another multiplexer 146. The controller 112 may comprise a control unit 152 coupled to a select logic/table 154. The combining

block 108 may comprise a summer. The boundary regulator 114 may comprise an adder/subtractor (ADS) 162.

[0026] In operation, the initial value selector/generator 102 may select an initial value from an initial vector ("VI") according to a control signal ("VI_SEL") 122 from controller 112. Alternatively, the initial value generator 136 of the initial value selector/generator 102 may generate an initial value as directed by the controller 112. The multiplexer 134 may select to output either a generated value from the generator 136 or a selected value from the multiplexer 132 in accordance with a control signal from the controller 112. The output of the initial value selector/generator 102 is input to the combining block 108.

[0027] As previously mentioned, the offset selector 104 may comprise multiplexers 142, 146 and one or more ACSs 144. The multiplexer 142 may select a value from a delta vector ("VD") according to a control signal 124 ("B1", "B0") from the controller 112. The selected delta vector ("VD") value may be input to the one or more ACSs 144, which may add or subtract an amount to the delta value. For example, the controller 112 may provide an initial value ("VI_ACS") and a subtract value ("SUB_V" or "SUBTRACT_V") to the ACSs 144. The VI_ACS value may provide a base value to which the VD value described above may be combined with (*i.e.*, added to). The SUB_V value may be subtracted from the combination of VD and VI_ACS. In at least some embodiments, the ACSs 144 may function according to the Algorithm 1 shown below.

Algorithm 1

```
accumulatorbase = VI_ACS;  
For each clock cycle,  
  If accumulatorACS >= SUBTRACT_V  
    outputACS = accumulatorACS - SUBTRACT_V;  
  Else  
    outputACS = accumulatorACS;  
  End if  
  accumulatorbase = accumulatorACS + VDselected.
```

[0028] As shown in Algorithm 1, the ACSs 144 may implement an initial base value (“accumulator_{base}”) equal to a parameter value (“VI_ACS”). For example, if VI_ACS equals two, the first accumulator value (“accumulator_{ACS}”) will equal two rather than zero. For each clock cycle, each ACS 144 may calculate a new accumulator_{ACS} value. As shown in Algorithm 1, if the current accumulator_{ACS} value is greater than or equal to a parameter value (“SUBTRACT_V”) the output (“output_{ACS}”) of an ACS 144 may be equal to accumulator_{ACS} - SUBTRACT_V. Otherwise, the output_{ACS} value may equal the accumulator_{ACS} value. If the accumulator_{base} value is equal to the combination of the accumulator_{ACS} value and the VD_{selected} value, the ACSs 144 have completed a cycle of interleaving (i.e., the output_{ACS} values will begin to repeat) for a particular interleaving algorithm. The multiplexer 146 may receive the outputs of the ACSs 144 and select/output a value according to a control signal from the controller 112. The output of the offset selector 104 may be combined with the output initial vector value by the combining block 108.

[0029] The pruning adjuster 106 may function with the controller 112 to control bit pruning and burst mapping as previously described. Accordingly, the pruning adjuster 106 may output a signal to the combining block 108, whereby the output

of the combining block 108 (the offset index value previously described) may be adjusted. As shown in Figure 2, the pruning adjuster 106 may function in accordance with a control signal 126 from the control unit 152 of the controller 112. In at least some embodiments, the pruning adjuster 106 may function to add, update and/or adjust values stored in the select logic/table 154 of the controller. In some embodiments, the system 101 may automatically account for bit pruning values based on a set of parameters. Therefore, no time (e.g., clock cycles) is wasted to discard unwanted bit index values.

[0030] As previously described, the output of the combining block 108 may be called an offset index value. The offset index value may be input to the boundary regulator 114, which may ensure the index value is within the index boundary as previously described. In operation, the ADS 162 of the boundary regulator 114 may receive the offset index value from the combining block 108 and a control signal from the control unit 152 of the controller 112. For example, the control signal may indicate whether the ADS 162 should add or subtract one or more pre-determined values. In at least some embodiments, the ADS 162 may function according to the Algorithm 2 shown below.

Algorithm 2

```

sumADS = VIselected + offsetselected + prune_valueselected;
If sumADS >= SUBTRACT_V
    outputADS = sumADS - SUBTRACT_V;
Else
    outputADS = sumADS.

```

[0031] As shown in algorithm 2, the ADS 162 may calculate a sum value ("sum_{ADS}") by summing a selected initial value ("VI_{selected}") with a selected offset value ("offset_{selected}") and a selected prune value ("prune_value_{selected}"). If sum_{ADS}

is greater than or equal to a parameter value "SUBTRACT_V" (also called "SUB_V"), then the output ("output_{ADS}") of the ADS 162 may equal $\text{sum}_{\text{ADS}} - \text{SUBTRACT_V}$. Otherwise, output_{ADS} may equal sum_{ADS} . The output of the ADS 162 may be used as an output value of the interleaver/de-interleaver 101.

[0032] The output of the interleaver/de-interleaver 101 may be input to an address calculator 116 as previously described. As shown in Figure 2, the address calculator 116 may combine a base address and the interleaver/de-interleaver output value (also called an offset index value) to output an interleaved or de-interleaved index of addresses. In some embodiments, the controller 112 may function in conjunction with the address calculator 116 to interleave/de-interleave a block a data to and/or from multiple blocks of base addresses.

[0033] In at least some embodiments, the systems 100, 101 may allocate bits of data to a burst such that one burst is completed before the next burst is created. When a burst has been completed, that burst may be transmitted. In contrast, other interleaver/de-interleavers follow a pattern in which all of the bursts receive a single bit of data before any of the bursts receive the second bit of data. The process of allocating one bit to each burst is typically repeated until all of the bursts are filled to capacity. Therefore, some embodiments of the invention may require less memory to buffer data bursts prior to transmission than others interleavers and/or de-interleavers that buffer all (or nearly all) data to bursts before transmitting.

[0034] As an example, consider a data block having 64 bits of data to interleave/de-interleave. More specifically, an interleaver may interleave the 64 bits according to the following pattern of bit index positions: [0, 30, 60, 20, 50, 10, 40, 5], [35, 15, 45, 25, 55, 3, 33, 63], [13, 43, 23, 53, 8, 38, 18, 48], [28, 58, 1, 31, 61, 11, 41, 21], [51, 6, 36, 16, 46, 26, 56, 4], [34, 14, 44, 24, 54, 19, 49, 9], [39, 29, 59, 12, 42, 2, 32, 63], [7, 37, 22, 52, 27, 57, 17, 47]. In some embodiments, the algorithm may separate the data into 8 bursts of data comprising 8 bits each such that the bursts of data are formed together a bit at a time (*i.e.* burst 0 receives a bit, burst 1 receives a bit, burst 2 receives a bit, etc.) until all of the bursts are filled.

[0035] As shown, burst 0 may comprise bits 0, 30, 60, 20, 50, 10, 40, and 5. Burst 1 may comprise bits 35, 15, 45, 25, 55, 3, 33, and 63. Burst 2 may comprise bits 13, 43, 23, 53, 8, 38, 18, and 48. Burst 3 may comprise bits 28, 58, 1, 31, 61, 11, 41, and 21. Burst 4 may comprise bits 51, 6, 36, 16, 46, 26, 56, and 4. Burst 5 may comprise bits 34, 14, 44, 24, 54, 19, 49, and 9. Burst 6 may comprise bits 39, 29, 59, 12, 42, 2, 32, and 63. Burst 7 may comprise bits 7, 37, 22, 52, 27, 57, 17, and 47.

[0036] As previously explained, instead of filling the bursts together one bit at a time as described by the interleaving algorithm, some embodiments of the invention may complete and transmit burst 0, then complete and transmit burst 1, etc., until all of the bursts have been completed and transmitted. Accordingly, less memory is required to buffer (temporarily store) 8 bits of data assigned to one burst as opposed to buffering approximately 64 bits of data as would be

required if the data is distributed to burst 0 through burst 7 in an alternating bit to burst allocation scheme (*i.e.* burst 0 receives a bit, burst 1 receives a bit, burst 2 receives a bit, etc.)

[0037] Figure 3 illustrates a table of parameters that may be used to implement the interleaver/de-interleaver 101 of Figure 2. Specifically, Figure 3 illustrates a set of general parameters that may be used to implement a variety of interleaving/de-interleaving algorithms (*i.e.* techniques) using the interleaver/de-interleaver 101. As shown, the parameters may comprise an initial vector ("VI"), an initial vector selection control ("VI_sel"), a delta (offset) vector ("VD"), an ACS initial value ("ACS_VI"), an ACS update rate, an adjust value ("Subtract_V"), a select line ("B0") for a VD multiplexer, another select line ("B1") for the VD multiplexer, a select line ("B2") for multiplexer 108, a number of address pointers value ("N_addr_ptr"), and a burst/code of blocks index calculation ("bst/cdbk index calculation"). Additionally, the table of Figure 3 also includes a "notes" section that is used to describe aspects of the interleaving/de-interleaving process.

[0038] The VI parameter may be input to initial value selector 102. As previously described the initial vector may be a vector of variable length. Additionally, VI may comprise a plurality of vectors. As shown in Figure 3, the VI for interleaving may comprise (0, 98, 82, 66) while the VI for de-interleaving may comprise four vectors: (0, 228), (57, 285), (114, 342), (171, 399). In at least some embodiments, only one vector is used at a time. For example, the vector (0, 228) may be used to assemble a burst "0" from one or more different code blocks

("CDBKs"). Specifically, the vector (0, 228) may be used to assemble burst "0" for CDBKs "0" and "-1." Based on the VI parameter, the system 101 may output a number of addresses based on code block pointers ("CDBK_ptr") plus an offset value. For example, if the vector (0, 228) is used as the VI parameter for burst "0," the system 101 may output the address locations for burst "0" as: CDBK0_ptr + 0, CDBK1_ptr + 228, CDBK0_ptr + 64, CDBK1_ptr + 292, CDBK0_ptr + 128, etc. In the address locations described above, an offset of 64 is added to the start locations "0" and "228" for the CDBL0 and CDBK1 addresses respectively. This process of adding 64 to an address is continued (for this particular algorithm) until the first burst is assembled. Once the first burst is assembled, the second VI vector (57, 285) may be used for the next burst and so on. This process is repeated. For this particular algorithm, eight bursts are assembled (after four bursts the CDBKs used are CDBK 1 and CDBK 0).

[0039] The VI_SEL parameter may be input to the initial value selector 102 as a control signal (e.g. signal 122) that permits the initial value selector 102 to select one of the VI values. As shown in Figure 3, VI_SEL may comprise a signal "K[B1B0]" (interleaving) or "J[B0]" (de-interleaving) that correspond to bit 1 "B1" and/or bit 0 "B0" values taken from a linear index counter J or K. As previously described, either K or J may be a linear index which may then be interleaved/de-interleaved. For example, in at least some embodiments K[B1B0] may follow a repeated two-bit pattern 00, 01, 10, 11 (i.e. a repeated 0, 1, 2, 3 pattern) for interleaving and J[B0] may follow a repeated one-bit pattern 0, 1 for de-

interleaving. The signals K[B1B0] and/or J[B0] may correspond to control signal 122 shown in Figures 1 and 2.

[0040] The VD parameter may be input to the offset selector 104. As previously described VD may be a vector of variable length. The ACS_VI parameter may be input to the ACSs 144 shown in Figure 2. The ACS_VI provides the ACSs 144 with an initial value. As shown in Figure 3, the ACS_VI may comprise “0” for both interleaving and de-interleaving.

[0041] The ACS update rate may control the ACSs 144 shown in Figure 2. In some embodiments, the ACS update rate controls how often an accumulator of each ACS 144 updates the VD value (described above). As shown in Figure 3, the ACS update rate may be “1/4” (*i.e.*, the ACSs 144 are updated every four cycles) for interleaving and “1/2” (*i.e.*, every two cycles) for de-interleaving. More specifically, if system 100 is used for de-interleaving, the ACSs 144 of offset selector 104 may start with an ACS initial value (ACS_VI) of “0” as previously described. If $VD = 64$, and the ACS update rate $= 1/2$ as previously described, the pattern followed by the output of the ACSs 144 would be 0, 0, 64, 64, 128, 128, etc. The value stored by each ACS 144 may be added to or subtracted from the selected offset value.

[0042] Eventually the offset value used by the ACSs 144 may force the index value to go beyond a desired index boundary. Accordingly, the adjust value (“SUB_V”) may be used to compensate for such situations. The SUB_V parameter may be input to the ACSs 144 and the boundary regulator 114 shown in Figure 2 to adjust the index value such that the index value is moved to within

a desired boundary. As shown Figure 3, SUB_V may equal “114” for interleaving and “456” for de-interleaving. For example, if the index boundaries are [0:455] as illustrated in algorithm 301 (shown in Figure 3), and the system 101 is used for de-interleaving then the amount of 456 may be subtracted from an index value whenever that index value is greater than 455.

[0043] “B0” (e.g. a multiplexer select line control) may be input to offset selector 104 as a control line. Accordingly, the offset selector 104 may select a VD value according to B0. As shown in Figure 3, B0 may equal a $K[B2]$ value, where $K[B2]$ is the second bit taken from a linear index K. Specifically, B0 may comprise a repeated 00001111 pattern for interleaving and constant “0” for de-interleaving. “B1” (e.g. a multiplexer select line control) also may be input to offset selector 104 as a control line. As shown in Figure 3, B1 may equal “0” for both interleaving and de-interleaving. B0 and B1 may be used to select the “VD” parameters illustrated of Figures 2.

[0044] The parameter B2 (e.g. a multiplexer select line control) may be input to multiplexing logic 108 as a control line. Therefore, the multiplexing logic 108 may select which ACS value to forward to computation block 110 according to B2. In some embodiments, B2 may equal “0” for both interleaving and de-interleaving. In such embodiments, only one of the ACS block is used. However in other embodiments interleaving algorithms may be more complicated and thus require more sophisticated control of multiplexing logic 108. Additionally, some embodiments may utilize additional (more than two) ACSs 144. In at least some

embodiments, the B2 value may comprise multiple bits for each clock cycle. For example, Figure 7D illustrates an embodiment in which B2 comprises two bits.

[0045] The number of address pointers (“NUM_ADDR_PTR”) parameter may be used by system 100 when implementing bursts as previously described. As shown in Figure 3, NUM_ADDR_PTR may equal eight for de-interleaving (*i.e.* one block of code is assembled from 8 bursts). Additionally, NUM_ADDR_PTR may equal two for interleaving (*i.e.* one burst is assembled from two blocks of code). NUM_ADDR_PTR may be input to controller 112, offset adjuster 106, and/or address calculator 116 in order to assemble bursts and blocks of code.

[0046] BST/CDBK index calculation may be performed in the controller 112 and output to the address calculator 116 such that the index locations of bursts and blocks of code may be determined. As shown in Figure 3, the BST/CDBK index calculation may be equal to $K[B2B1B0]$ (*i.e.* $\text{MOD}(K,8)$) for an interleaver and $N - \text{MOD}(J,2) + \text{FLOOR}(\text{BST_IDX}/4)$ for a de-interleaver. In the function “ $N - \text{MOD}(J,2) + \text{FLOOR}(\text{BST_IDX}/4)$ ”, BST_IDX corresponds to burst indices 0, 1, 2, 3, 4, 5, 6, and 7, N corresponds to the index of a block of code, MOD corresponds to a modulus function where “ $\text{MOD}(J, 2)$ ” = $J - 2 * \text{FLOOR}(J/2)$, and FLOOR corresponds to rounding to the next lowest integer.

[0047] The parameters illustrated in Figure 3 may be used for many different interleaving/de-interleaving algorithms as will later be described. By using the parameter values shown in the table of Figure 3, the system 101 may interleaving or de-interleaving (reverse interleaving) according to the algorithm 301. The algorithm 301 may be found in section 3.1.3 of the GSM 05.03 V8.5.0

release 1999 standard. By changing and/or programming the parameter values, the system 101 may function as an interleaver/de-interleaver for many different algorithms. These algorithms may be found in standards such as GSM/GPRS/EDGE, WCDMA, and IS2000.

[0048] Figures 4A-4B illustrate a software implementation of an interleaving/de-interleaving algorithm. As shown in Figure 4A, the algorithm 401 found in section 3.1.3 of the GSM 05.03 V8.5.0 release 1999 standard may be implemented. The Figure 4A illustrates an embodiment of “hardware code” 403 which illustrates the functionality of the interleaver/de-interleaver 101. Figure 4B illustrates hardware code related to several interleaving/de-interleaving patterns illustrated in columns 0-7 of Table 1 of the GSM 3.1.3 standard.

[0049] Figures 5A-5S illustrate tables of parameters and parameter values that may be used to implement interleaving and/or de-interleaving algorithms found in the GSM 05.03 V8.5.0 release 1999 standard using the interleaver/de-interleaver 101 of Figure 2. Specifically, the parameters and parameter values permit system 101 to execute interleaving and/or de-interleaving as described by various tables and algorithms found in the GSM 05.03 V8.5.0 release 1999 standard. In particular, the system 101 may implement MCS-5 EDGE interleaving/de-interleaving algorithms (e.g., as shown in Figures 5K and 5L) without using the large look-up table provided by the GSM standard. Specifically, the GSM standard provided the look-up tables for certain algorithms due to a lack of a closed-form description for the interleaving algorithm. Accordingly, embodiments of the invention may implement closed-form versions of these

GSM algorithms using the hardware and parameters described previously for Figures 1 and 2. While embodiments of the invention may be used to implement interleaving algorithms found, for example, in industry standards such as the GSM 05.03 V8.5.0 release 1999 standard, the invention is not limited to any particular standard.

[0050] Figures 6A and 6B illustrate an interleaving algorithm according to a WCDMA (Wideband Code Division Multiple Access) standard found in the 3GPP TS 25.212-v.3.5.0 (2000-12) release 1999 standard. Figures 6C-6D illustrate a table of parameters and parameter values that may be used with the system 101 to implement the algorithm illustrated in Figures 6A and 6B. Figures 6E and 6F illustrate a 30x30 look up table (LUT) that may be used by the system 101 to provide a de-pruning adjustment when implementing the WCDMA standard illustrated in Figures 6A and 6B. Specifically, the LUT contains adjustment values that may be used by the boundary regulator 114 to adjust the combined initial value and offset value such that the extraneous (dummy) bits (referred to in Fig. 6B) are automatically “pruned” without using extra logic and/or waiting cycles.

[0051] Figures 7A-7E illustrate tables of parameters and parameter values that may be used to implement a number of interleaving and/or de-interleaving algorithms found in the IS2000 standard, referenced in 3GPP2 C.S0002-C VERSION 1.0.

[0052] Figure 8 illustrates an embodiment of an apparatus 190 that may implement the interleaving/de-interleaving systems 100 and/or 101 shown in

Figures 1 and 2. As shown in Figure 8, the apparatus 190 may comprise processing unit 192 coupled to a memory 194 and a transceiver 196, wherein the transceiver implement the interleaver/de-interleaver 101. The apparatus 190 may be representative of a cell phone, personal digital assistance, laptop computer, or any other device that may use interleaving/de-interleaving when communicating. Additionally, embodiments of the invention may be implemented with any transmitter and/or receiver of a communication system.

[0053] Figure 9 illustrates a method 900 for interleaving/de-interleaving a block of data in accordance with embodiments of the invention. As shown in Figure 9, the method 900 may comprise receiving an initial value (block 902) and receiving an offset value (block 904). As previously explained, the initial value and offset values may be selected from one or more vectors. Alternatively, the initial value and/or offset value may be generated as previously described. If the offset needs to be adjusted as determined at block 906, the offset may be adjusted accordingly (block 908), then combined with the initial value (block 910). As described above, the offset may be adjusted to account for bursts and/or bit pruning. If the offset does not need to be adjusted as determined at block 906, the unadjusted offset value is combined with the initial value (block 910).

[0054] At block 912, a determination is made as to whether the combined initial value and offset is within a pre-determined boundary (e.g. an index boundary). If the combined value is not within the pre-determined boundary as determined at block 912, the combined value may be adjusted so that it is moved to within the pre-determined boundary (block 914). The adjusted combined value may then

be output as an index value (block 916). If the combined value is determined to be within the pre-determined boundary (block 912), the combined value may be output as an index value (block 916). As described above, the output index value may be used by an address calculator or other hardware or software functions to interleave and/or de-interleave a block of data.

[0055] While the preferred embodiments of the present invention have been shown and described, modifications thereof can be made by one skilled in the art without departing from the spirit and teachings of the invention. For example, some embodiments may implement other existing interleaving algorithms that were not mentioned, or future interleaving standards. The embodiments described herein are exemplary only, and are not intended to be limiting. Many variations and modifications of the invention disclosed herein are possible and are within the scope of the invention. Accordingly, the scope of protection is not limited by the description set out above. Each and every claim is incorporated into the specification as an embodiment of the present invention.